CLAIMS:

What is claimed is:

1. A method, in a data processing system, for code reusability and maintainability, the method comprising:

providing a utility class in a server that defines a utility method;

responsive to receiving a request at the server for attributes for an entity from a client, generating a method call for the utility method, wherein the method call identifies the entity and a response object name;

generating a response object and assigning the response object name to the response object; and

returning the response object to the client.

2. The method of claim 1, wherein the utility method is a Java public static method.

3. The method of claim 1, wherein the request is an extensible markup language request.

4. The method of claim 3, wherein the extensible markup language request is one of a list request and a get request.

5. The method of claim 1, further comprising:

retrieving, by the utility method, at least one data item for the method call and the identified entity,

wherein the response object include the at least one data item.

6. The method of claim 5, wherein the step of retrieving at least one data item includes retrieving the at least one data item from a database.

7. The method of claim 6, wherein the at least one data item is retrieved from the database through a structured query language interface.

8. The method of claim 5, wherein the request includes a list of attributes.

9. The method of claim 8, wherein the at least one data item includes a set of attributes for the entity, wherein the set of attributes corresponds to the list of attributes.

10. The method of claim 9, wherein the list of attributes is an empty string.

11. The method of claim 10, wherein the set of attributes includes all attributes for the entity.

12. The method of claim 1, wherein the response object is an extensible markup language document.

13. An apparatus, in a data processing system, for code reusability and maintainability, the apparatus comprising:

a utility class that defines a utility method;

a program interface, wherein the program interface, responsive to receiving a request for attributes for an entity from a client, generates a method call for the utility method, wherein the method call identifies the entity and a response object name;

wherein the program interface generates a response object and assigns the response object name to the response object; and

wherein the program interface returns the response object to the client.

14. The apparatus of claim 13, wherein the client includes an extensible markup language interface and wherein the request is an extensible markup language request.

15. The apparatus of claim 13, wherein the utility method retrieves at least one data item for the method call and the identified entity and wherein the response object include the at least one data item.

16. The apparatus of claim 15, wherein the utility method retrieves the at least one data item from a database.

17. The apparatus of claim 15, wherein the request includes a list of attributes.

18. The apparatus of claim 17, wherein the at least one data item includes a set of attributes for the entity,

wherein the set of attributes corresponds to the list of attributes.

19. The apparatus of claim 18, wherein the list of attributes is an empty string.

20. The apparatus of claim 19, wherein the set of attributes includes all attributes for the entity.

21. The apparatus of claim 13, wherein the response object is an extensible markup language document.

22. A computer program product, in a computer readable medium, for code reusability and maintainability, the computer program product comprising:

instructions, in a utility class, for defining a utility method;

instructions, responsive to receiving a request at the server for attributes for an entity from a client, for generating a method call for the utility method, wherein the method call identifies the entity and a response object name;

instructions for generating a response object and assigning the response object name to the response object; and

instructions for returning the response object to the client.